Amendments to the Claims

1. (Currently Amended) A method for ~~capturing one or more graphic primitives and attributes associated with such graphic primitives of a display object~~ a calling process to manipulate information presented by a target process, the method comprising ~~the steps of~~:

injecting a spy component into [[a]] the target process by [[a]] the calling process, wherein the spy component is an executable program module;

capturing [[the]] one or more ~~graphic~~ graphics primitives and attributes associated with such ~~graphic~~ graphics primitives during [[the]] execution of the target process; [[and]]

returning the ~~graphic~~ graphics primitives and attributes associated with such ~~graphic~~ graphics primitives to the calling process; and

producing, by the calling process, an output, the output based, at least in part, on the returned graphics primitives and attributes associated with such graphics primitives.

2. (Original) The method of claim 1, wherein the step of injecting the spy component is performed by an injection component that is invoked by the calling process.

3. (Original) The method of claim 2, wherein the injection component is an executable program module comprising executable routines for injecting source code into a target process.

4. (Original) The method of claim 1, wherein the step of injecting includes inserting one or more function patches into one or more executable program modules that correspond to an operating system upon which the target process is being executed, the executable program modules having instructions for rendering graphics primitives to a graphical user interface.

5. (Original) The method of claim 4, wherein the step of inserting the function patches is performed by the spy component.

6. (Original) The method of claim 1, wherein the step of injecting includes installing one or more hook functions into the operating system APIs that generate system messages in the event of a display object being output to a user interface screen during the execution of the target process.

7. (Original) The method of claim 6, wherein the step of installing is performed by the spy component.

8. (Original) The method of claim 6, wherein the system messages provide context information that is descriptive of an invoked action within the target process.

9. (Original) The method of claim 6, wherein the one or more hook functions are installed by a hook management component that is called upon by the spy component after injection into the target process.

10. (Original) The method of claim 9, wherein the hook management component is responsible for uninstalling the one or more hook functions upon termination of the target process.

11. (Currently Amended) The method of claim [[1]] 6, wherein the step of capturing includes calling the one or more hook functions to intercept any system messages generated as a result of an invoked action within the target process.

12. (Currently Amended) The method of claim 11, wherein the one or more hook functions set a flag to activate [[the]] one or more function patches installed by the spy component.

13. (Currently Amended) The method of claim 1, wherein the step of capturing includes invalidating a display object that is output to a user interface as a result of [[the]] an invoked action within the target process.

14. (Currently Amended) The method of claim 13, wherein the step of invalidating includes calling [[the]] function patches to capture the graphics primitives and attributes associated with such graphics primitives as they are drawn to the user interface to render the display object.

15. (Currently Amended) The method of claim 1, wherein the step of returning includes sending [[the]] context information captured by [[the]] one or more hook functions to the calling process as a system message.

16. (Currently Amended) The method of claim 1, wherein the step of returning further includes sending the graphics primitives and attributes associated with such graphics primitives that are captured ~~by the one or more function patches~~ to the calling process as a system message.

17. (Currently Amended) A computer-readable medium having computer-executable instructions for ~~capturing one or more graphic primitives and attributes associated with such graphic primitives of a display object~~ a calling process to manipulate information presented by a target process, the computer-executable instructions performing steps comprising:

injecting a spy component into [[a]] the target process by [[a]] the calling process, wherein the spy component is an executable program module;

capturing [[the]] one or more ~~graphic~~ graphics primitives and attributes associated with such ~~graphic~~ graphics primitives during [[the]] execution of the target process; [[and]]

returning the ~~graphic~~ graphics primitives and attributes associated with such ~~graphic~~ graphics primitives to the calling process; and

producing, by the calling process, an output, the output based, at least in part, on the returned graphics primitives and attributes associated with such graphics primitives.

18. (Currently Amended) A system for ~~capturing one or more graphic primitives and attributes associated with such graphic primitives of a display object~~ manipulating information presented by a target process, the system comprising:

an injection component for injecting a spy component into [[a]] the target process residing on a computer;

a spy component for capturing graphics primitives and attributes associated with such graphics primitives in connection with system messages that are generated by the target process as a result of an invoked action within the target process; [[and]]

a hook management component for installing and uninstalling one or more hook functions into one or more program modules that are executed by an operating system residing on the computer, the program modules having instructions for generating system messages during the execution of the target process; and

a calling process for producing an output based, at least in part, on the captured graphics primitives and attributes associated with such graphics primitives.

19. (Original) The system of claim 18, wherein the injection component is an executable program module consisting of executable instructions for injecting the spy component into the executable code of the target process.

20. (Currently Amended) The system of claim 18, wherein the injection component injects the spy component into the target process on behalf of [[a]] the calling process.

21. (Currently Amended) The system of claim 20, wherein the calling process is a ~~computer executable~~ computer-executable application.

22. (Original) The system of claim 18, wherein the spy component inserts one or more function patches into one or more executable program modules that correspond to the operating system upon which the target process is being executed, the executable program modules having instructions for rendering graphics primitives to a graphical user interface.

23. (Original) The system of claim 22, wherein the function patches capture graphics primitives and associated attributes of such graphics primitives that are rendered to a user interface by a display object as a result of an action invoked within the target process.

24. (Currently Amended) The system of claim 23, wherein the spy component packages the graphics primitives and attributes associated with such graphics primitives and sends [[it]] them to the calling process as a system message.

25. (Currently Amended) The system of claim 18, wherein the spy component calls the hook management component to insert one or more hook functions into one or more executable program modules that correspond to the operating system upon which the target process is being executed, the executable program modules having instructions for generating system messages in the event of a display object being output to a user interface during [[the]] execution of the target process.

26. (Currently Amended) The system of claim 25, wherein the one or more hook functions set a flag to activate [[the]] one or more function patches installed by the spy component.

27. (Original) The system of claim 25, wherein the one or more hook functions intercept any system messages generated as a result of an invoked action within the target process.

28. (Original) The method of claim 27, wherein the system messages contain context information that is descriptive of an invoked action within the target process.

29. (Currently Amended) The system of claim 18, wherein the spy component packages [[the]] context information and sends it to the calling process as an OS message.

30. (Original) The system of claim 18, wherein the hook management component is responsible for installing and uninstalling hook functions on behalf of the spy component.

31. (Original) The system of claim 30, wherein the hook functions are uninstalled by the hook management component upon termination of the target process.

32. (New) The method of claim 1, wherein the calling process is selected from the group consisting of: a localization tool, a language-processing application, a text-to-speech application, and an operating system utility.

33. (New) The method of claim 1, wherein producing an output comprises performing an operation selected from the group consisting of: displaying an image on a display screen and producing an audio output.

34. (New) The system of claim 18, wherein the calling process is selected from the group consisting of: a localization tool, a language-processing application, a text-to-speech application, and an operating system utility.

35. (New) The system of claim 18, wherein producing an output comprises performing an operation selected from the group consisting of: displaying an image on a display screen and producing an audio output.